MCGINN & GIBB, P.C.
A PROFESSIONAL LIMITED LIABILITY COMPANY
PATENTS, TRADEMARKS, COPYRIGHTS, AND INTELLECTUAL PROPERTY LAW
1701 CLARENDON BOULEVARD, SUITE 100
ARLINGTON, VIRGINIA 22209
TELEPHONE (703) 294-6699
FACSIMILE (703) 294-6696

APPLICATION FOR UNITED STATES LETTERS PATENT

APPLICANT: Shusaku Uchibori

FOR: TASK PROCESSING SYSTEM

DOCKET NO.: F-10190

TASK PROCESSING SYSTEM

BACKGROUND OF THE INVENTION

1. Technical Field of the Invention

The present invention relates to a task processing system, and particularly, to a task processing system for speeding up task processing by controlling generation of overhead for task switching in the case of processing a large number of small sized- tasks.

2. Description of the Prior Art

In a conventional task processing system, when the tasks to be processed are known in advance, the task processing speed of the system is optimized by scheduling the tasks beforehand. On the other hand, when it is impossible to schedule the tasks, the tasks are processed as asynchronous events such as interrupts. For example, in the "Queue Processing Method" as disclosed in JP10-326197 A (1998), ordinary events from software and interrupt events caused by a timer or hardware interrupt are accepted at any time, and are registered into an input key. Then, they are executed sequentially without prohibiting interrupt.

As shown in Fig. 5, in the above-mentioned "Queue Processing Method", the queue entries of ordinary events and interrupt events are prepared. When interrupt 1 event occurs, its content is stored in interrupt 1 event information block, and its event queue is added to the interrupt 1 event queue. Interrupt 2 event is processed in the same way. The data stored in the interrupt 1 event information block and the

interrupt 2 event information block is registered again as the ordinary event, by releasing the registration of the interrupt event queue without transferring it to the ordinary event information block.

However, the above-mentioned conventional task processing system has a disadvantage that it cannot improve the speed of the task processing of the whole system, because it is impossible to schedule the tasks beforehand. Therefore, it becomes necessary to generate overheads for acquiring and releasing the resources necessary for the task processing.

The conventional task processing system has another disadvantage that it is not suitable for processing a large number of small-sized tasks, because a lot of overheads for task switching becomes necessary.

SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to speed up the task processing by controlling the overhead generation for task switching in the case of processing a large number of small-sized tasks.

The task processing system of the present invention comprises a storage means for storing an identifier of a generated event, a task control device for creating a task based on the above event, and a task processing device for processing the task. The task processing device searches the identifier for creating the same task as the processed task, so as to further process the same task as the processed task.

The task processing system of the present invention

processes a task depending on the kind of a created event. Concretely, it checks whether or not an event of the same kind occurred, after completion of the task processing. When an event of the same kind has occurred, the same task is continuously processed. Therefore, the overhead for acquiring and releasing resources necessary for the task processing is reduced, thereby speeding up the task processing on the whole system.

According to the present invention, it is possible to speed up the task processing in the whole system, because the overheads for acquiring and releasing a resource necessary for the task processing are reduced.

Further, according to the present invention, it is possible to process a large number of small-sized tasks at higher speed, because the tasks are switched at higher speed, due to the overhead reduction.

BRIEF EXPLANATION OF THE DRAWINGS

Fig. 1 is a block diagram of the task processing system of the first embodiment of the present invention.

Fig. 2 is a flow chart for describing the operation of the task processing system as shown in Fig.1.

Fig. 3 is a block diagram of the task processing system of the second embodiment of the present invention.

Fig. 4 is a flow chart for describing the operation of the task processing system as shown in Fig. 3.

Fig. 5 is a sequence diagram for explaining a conventional queue processing method.

PREFERRED EMBODIMENT OF THE INVENTION

A block diagram of the task processing system of the present invention is shown in Fig.1. The task processing system as shown in Fig.1 comprises event creating device 1 such as an interrupt creating device, storage means 2 for storing information, task processing device 4 operated by a program control, and task control device 3 for controlling task processing device 4, depending on the created event.

Storage means 2 comprises event storing unit 21 for storing the events created by event creating device 1, and task resource storing unit 22 for storing the resources necessary for the task processing of programs and data. An event stored in event storing unit 21 includes event identifier 211 for identifying the kind of the event.

Task control device 3 comprises event check means 31, task creating means 32, and task resource management means 33.

When task processing device 4 completes the task processing, event check means 31 checks whether the event for creating the same task as the completed task is registered in the event storing unit 21. Event identifier 211 is used for the check.

Task creating means 32 controls task processing device 4 so as to activate a task corresponding to the event registered in event storing unit 21. Event identifier 211 identifies the corresponding task.

Task resource management means 33 acquires a resource at the time of starting the task, and releases the resource at the time of completing the task.

The task corresponding to the event registered in event storing unit 21 is created by task creating means 32, and processed by task processing device 4. The resource necessary for the task processing is read out from task resource storing unit 22 by task resource management means 33, and supplied to the task processing device 4. Thus, the necessary resource is acquired.

At the time when the task processing has been completed in task processing device 4, event check means 31 checks whether an event of the same kind as that of the event having created the completed task is registered in event storing unit 21. When it is not registered, a resource required to be rewritten to the task resource storing unit 22, among the resources required by the completed task, is returned back to task resource storing unit 22, through the task resource management means 33. Thus, the resource is released.

When an event of the other kinds is registered in event storing unit 21, the task creation and resource acquisition are repeated. When an event of the same kind as that of the event having created the completed task is registered in event storing unit 21, the same task is continuously processed by task processing device 4 without releasing the resource.

Thus, sequential execution of the same task helps to reduce the overhead for the release and acquisition of resources, when various tasks are processed, thereby speeding up the task processing on the whole system.

Fig. 2 is a flow chart for describing the operation of the task processing system of the present invention. An event created by event creating device 1 is stored in event storing unit 21.

Event identifier 211 indicating the kind of the event is also stored in event storing unit 21.

Event check means 31 checks whether an event having a specified identifier is stored in event storing unit 21 (Steps A1 and A2).

When the event having the specified identifier is stored, task resource management means 33 reads out a resource necessary for the task corresponding to event identifier 211 from task resource storing unit 22 and feeds the read out resource to the task processing device 4 (Step A3).

Further, task creating means 32 activates or creates a task corresponding to event identifier 211, by using task processing device 4, while the corresponding event is deleted from event storing unit 21 (Step A4).

Event check means 31 checks whether an event having the same identifier as event identifier 211 corresponding to the completed task is stored in event storing unit 21 (Steps A5 and A6).

When an event having the same identifier is stored, the step returns to Step A4, where the same task as the completed task is created by task creating means 32, and the corresponding event is deleted from event storing unit 21.

Finally, when an event having the same identifier is not stored, task resource management means 33 rewrites, into task resource storing unit 22, the resource which has been used by the completed task and should be released (Step A7).

A block diagram of another embodiment of the task processing system of the present invention is shown in Fig.3. The embodiment as shown in Fig. 3 is different from the

embodiment as shown in Fig. 1 in that a plurality of task processing devices 41, 42, 4N are provided, in place of task processing device 4, and further in that storage means further comprises executing task storing unit 23.

Executing task storing unit 23 stores each task which is being executed by task processing devices 41 to 4N. The task stored by executing task storing unit 23 also includes event identifier 231 for identifying the kind of the task.

A flow chart for explaining the operation of the task processing system as shown in Fig. 3 is shown in Fig.4. The flow charts are identical, and operating in parallel for the number N of task processing devices.

The operations of event check means 31, task creating means 32, and task resource management means 33 in this embodiment as shown from Step B31 to Step B61 in Fig. 4 are the same as those of respective means 31, 32, and 33 in the embodiment shown in Fig. 1.

The event created by event creating device 1 is stored in event storing unit 21. Event identifier 211 indicating the kind of the event is also stored in event storing unit 21.

Event check means 31 checks whether any other event having an identifier other than event identifier 231 of the task stored in executing task storing unit 23 is stored in event storing unit 21 (Steps B11 and B21).

In Step B41, task creating means 32 activates or creates a task corresponding to event identifier 211, by using task processing device 4, while the corresponding event is deleted from event storing unit 21 and simultaneously stored in executing task storing unit 23.

Finally, when no event having the same identifier is stored, in Step B71, task resource management means 33 rewrites, into task resource storing unit 22, a resource which have been used by the completed task and should be released. The corresponding task is deleted from the executing task storing unit 23.